ECE 241 – HOMEWORK 3 Fall 2021

Due: Thursday, October 7, 2021 at 11:00 PM on Gradescope

Q1 (50 points). In this question, we ask you to implement a hash table and use the chaining technique to handle collisions. The hash table is maintained in ***self.slots***. Each element in the ***self.slots*** stores the collisions as an OrderedList, where the list is sorted in ascending order of the keys. The nodes in the OrderedList are (key, data) pairs that are stored in a Node object.

You are asked to implement the ***put*** function in the HashTable class to insert a (key, data) pair into the hash table.
When putting a (key, data) pair to the hash table, the following rules should be applied:

1. Compute the hash value based on the input key which will be the position in the hash table.
2. When there is no collision, the position at the hash table is empty. Create a new OrderedList at that position and put the (key, data) pair into the OrderedList.
3. When there is a collision,
    a. If the key exists in the OrderedList, update the value with the new data.
    b. Otherwise, insert the (key, data) pair to the OrderedList

Q2 (50 points). Find the shortest path between 2 nodes in a binary search tree (balancing isn't implemented in this one). Refer the comments in the question2.py to modify function _find_path() which is called from find_path(). Follow instructions in_find_path() method to return list_path and steps.